

Artificial Intelligence Systems Xerox Lisp

**Index to Xerox Common Lisp
Implementation Notes
Lyric Release**

**INDEX TO XEROX COMMON LISP
IMPLEMENTATION NOTES**

XEROX

3103178
Lyric Release
June 1987

INDEX TO XEROX COMMON LISP IMPLEMENTATION NOTES

3103178

Lyric Release

June 1987

The information in this document is subject to change without notice and should not be construed as a commitment by Xerox Corporation. While every effort has been made to ensure the accuracy of this document, Xerox Corporation assumes no responsibility for any errors that may appear.

Copyright © 1987 by Xerox Corporation.

Xerox Common Lisp is a trademark.

All rights reserved.

"Copyright protection claimed includes all forms and matters of copyrightable material and information now allowed by statutory or judicial law or hereinafter granted, including, without limitation, material generated from the software programs which are displayed on the screen, such as icons, screen display looks, etc."

This manual is set in Modern typeface with text written and formatted on Xerox Artificial Intelligence workstations. Xerox laser printers were used to produce text masters.

A

- access functions 54
- accessors 57
- adjoin 43
- adjust-array (*Function*) 49
 - arguments 50
- advising 106
 - functions 109
- ambiguities 58-59, 61
- and (*Macro*) 5
- APPLY (array inspector command) 52
- aref 46-47
- array inspector 51
- array-dimension-limit 46
- array-rank-limit 46
- arrays 45-52
 - empty 47
 - extendable 45
 - degenerate 46
 - displaced 45
 - limits 46
 - read-only 45
- array-total-size-limit 46
- ASCII control characters 40
- assert (*Macro*) 72

B

- backquote 59
- backslash 117
- backtrace 117
- backward compatibility
 - 102, 104, 107, 109, 111
- bound handler 64
- break (*Function*) 74
- breaking 101

- breakpoints 112-114
- break-when expressions 101
- bt (debugger command) 117
- bt! (debugger command) 117
- btv (debugger command) 117
- btv! (debugger command) 117

C

- call-arguments-limit 7
- case (*Macro*) 7
- cautions 58-59
- ccase (*Macro*) 72
- cerror (*Function*) 71
- char-bits-limit 39
- char-code-limit 39
- char-font-limit 39
- character sets 40
- characters 39
 - control-bit functions 41
 - conversions 40
- check-type (*Macro*) 72
- cl:gensym 15
- clear-output 60
- close 58
- code template 108
- coerce (*Function*) 1
- comma 59
- compile (*Function*) 89
- compile-file (*Function*) 61, 89
- compiled functions 3
- compiler 89
- compiler:context-predicate-p
 - (*Function*) 95

- compiler:context-top-level-p
 (Function) 95
- compiler:context-values-used
 (Function) 95
- compiler:copy-env-with-fn
 (Function) 97
- compiler:copy-env-with-var
 (Function) 97
- compiler:env-boundp (Function) 96
- compiler:env-fboundp (Function) 96
- compiler:make-empty-env (Function)
 97
- compiler:optimize-and-macroexpand
 (Function) 97
- compiler:optimize-and-macroexpand
 -1 (Function) 97
- computation of forms 75
- condition 63
- conditions
 - defining 66
 - handling 72
 - signalling 70
- condition type hierarchy 83
- condition-signalling forms 70
- conditionals 7
- CONTROL-D 113
- CONTROL-E 113
- ctypcase (Macro) 72
- D**
- dbt (debugger command) 117
- dbt! (debugger command) 117
- debugger 112-122
 - breakpoints 101
 - errors 121
 - interface to 120
- debugger window 113
- debugging the compiler 89
- debugging tools 98
- declarations 13-14
 - specifiers 13
 - syntax 13
- declaring global variables 4
- default structure type 53
- defining optimizers
 - see xcl:deftoptimizer 93
- defmacro 7, 9, 92
- defpackage (NLambda Function) 19
- defparameter 4
- defstruct 5, 54
- defun 3, 9, 92
- DFASL 89
- displaced arrays
 - see arrays
- dlap 90
- do-all-symbols (Macro) 17
- do-external-symbols (Macro) 17
- do-symbols (Macro) 17
- documentation 98
- E**
- ecase (Macro) 72
- empty structures 53
 - see also structures
- equalp 5
- error (Condition) 85
- error (Function) 70
- error system 63-88
 - examples 74, 77-79

- terminology 63-66
- errors 64
- etypcase (*Macro*) 72
- eval (debugger command) 112, 114
- exporting symbols 31
- extensions
 - default file 61
 - LAP code 90
- extensions to standard packages 17
- F
 - fat characters 45
 - NS characters 58
 - file manager 61
 - file system interface
 - limitations 61
 - files
 - binary 61
 - extensions 61
 - text 61
 - finish-output 60
 - foo 1
 - functions
 - modifying behavior of 106
- G
 - graphic characters 40
 - graphic-char-p 40
- H
 - handler 65
 - handlers 63
 - hash colon 18
- I
 - IF-THEN-ELSE 91
 - il:* 107
 - il:*short-backtrace-filter* (*Variable*) 117
 - il:advise (file manager command) 111
 - il:advise (file manager command) 111
 - il:advise (Function) 109
 - il:autobacktraceflg (*Variable*) 120
 - il:backtracefont (*Variable*) 121
 - il:break (*NLambda Function*) 103
 - il:break0 (Function) 102
 - il:breakdelimiter (*Variable*) 116
 - il:breakregionspec (*Variable*) 121
 - il:bytemacro 9
 - il:characternames 40
 - il:compileuserfn (*Variable*) 91
 - il:compset (Function) 91
 - il:dassem.savelocalvars (Function) 91
 - il:dmacro 9
 - il:dwimify (Function) 92
 - il:dwimifycompflg (*Variable*) 92
 - il:equalall 5
 - il:gensym 15
 - il:helpclock (*Variable*) 120
 - il:helpdepth (*Variable*) 120
 - il:helpflag (*Variable*) 119
 - il:helptime (*Variable*) 120
 - il:ignoremacro 94
 - il:lastpos (*Variable*) 116
 - il:macro 9
 - il:nobind 3
 - il:readvise (*NLambda Function*) 110
 - il:rebreak (*NLambda Function*) 103
 - il:storage (Function) 122

il:unadvise (*NLambda Function*) 110
il:unbreak (*NLambda Function*) 103
il:unbreak0 (*Function*) 103
il:|MaxBkMenuWidth| (*Variable*) 120
il:|MaxBkMenuHeight| (*Variable*) 120
implementation parameters 37
inline declarations 13
INTERLISP (standard package) 17
Interlisp compiler 91
Interlisp macro 9

K

KEYWORD (standard package) 17
keywords
 see : (leading colon)
KOTO CML files 25

L

lambda expressions 3
lambda-parameters-limit 3
LAP code 90
lexical environments 92,96,99
limitations 61
LISP (standard package) 17
Lisp Assembly Program (LAP) 90
lists 43
load 61

M

make-array (*Function*) 46
 arguments to 45
make-context (*Function*) 95
make-package (*Function*) 18
Meta key 39

Meta-A 40
modules 20
multiple-values-limit 8

N

named constants 4,37
 limits 37
nlambda functions 11
non-interactive handling 77

O

ok (debugger command)
optimizer examples 95
or (*Macro*) 5

P

packages 17-36
 cautions 33
 changing 32
 defining 30-32
 error conditions 20-25
 IL package 34
 inheritance 33
 Interlisp-inheriting 35
 specifying 31
 Koto reader compatibility 25-32
 moving existing code into 29
 standard 17
 symbols 33-35
 exporting 31
parse-body (*Macro*) 9
pb (exec command) 116
pr (debugger command) 115
predefined proceed function 82
predicates 5,57
print functions 53-55

- printing syntax 55
- proceed 63
- proceed (debugger command) 115
- proceed case examples 81
- proceed case object 81
- proceed cases 75
- proceed function 65
- pushnew 43
- R**
- reader 59
 - compatibility feature 25-37
- recursive debugger entry 122
- recursive functions 3
- rename-package (*Function*) 19
- return &optional (debugger command) 115
- revert (debugger command) 119
- room (*Function*) 122
- S**
- semi-colon 61
- serious conditions 64
- setf 81
- setting breakpoints 101
- sharing structure 59
- SHOW (array inspector command) 51
- signal 63
- special variables 3
- standard packages 17
 - extensions to 17-20
- step (*Macro*) 112
- stepping 111
- stop (debugger command) 115
- stream functions 57, 58
- streams
 - ambiguities 58
 - cautions 58
 - subtypep (*Function*) 5, 54
 - svref 48
 - symbol print names 15
 - symbol-name 15
 - SYSTEM (standard package) 17
- T**
- the (special form) 1
- thin characters 45
- time (exec command) 122
- time (*Function*) 122
- trace (*Macro*) 104
- tracing 104
- transfer to a clause 75
- trees 43
- turn off tracing 104
- type-checking 53
- typep 5
- U**
- ub (debugger command) 114
- unsigned-byte 61
- untrace (*Macro*) 104
- USER (standard package) 17
- V**
- value (debugger command) 114
- W**
- warn (*Function*) 71
- wrappers facility 98
- X**
- XCL (standard package) 17

- XCL Compiler 91
- XCL-USER (standard package) 17
- xcl:*current-condition* (Variable) 70
- xcl:*print-structure* (Variable) 55
- xcl:*total-packages-limit* (Constant) 17
- xcl:*trace-length* (Variable) 105
- xcl:*trace-level* (Variable) 105
- xcl:*trace-verbose* (Variable) 105
- xcl:abort (Function) 82
- xcl:advise-function (Function) 109
- xcl:break-function (Function) 102
- xcl:broadcast-stream-p (Function) 57
- xcl:broadcast-stream-streams (Function) 57
- xcl:catch-abort (Macro) 81
- xcl:cell-error (Condition) 87
- xcl:compute-proceed-cases (Function) 80
- xcl:concatenated-stream-p (Function) 57
- xcl:concatenated-stream-streams (Function) 57
- xcl:condition (Condition) 64, 85
- xcl:condition-case (Macro) 72
- xcl:condition-handler (Macro) 69
- xcl:condition-reporter (Macro) 69
- xcl:control-error (Condition) 86
- xcl:create-trace-window (Function) 105
- xcl:debug (Function) 74
- xcl:debugger (Function) 121
- xcl:def-define-type (Function) 98
- xcl:default-proceed-report (Macro) 80
- xcl:default-proceed-test (Macro) 80
- xcl:defglobalparameter (Macro) 4
- xcl:defglobalvar (Macro) 4
- xcl:define-condition (Macro) 66
- xcl:define-proceed-function (Macro) 78
- xcl:defineinline (Macro) 3,13
- xcl:deftoptimizer (Macro) 3,93
- xcl:delete-package (Function) 18
- xcl:destructuring-bind (Macro) 7
- xcl:do-internal-symbols (Macro) 18
- xcl:do-local-symbols (Macro) 18
- xcl:echo-stream-input-stream (Function) 58
- xcl:echo-stream-output-stream (Function) 58
- xcl:echo-stream-p (Function) 57
- xcl:end-of-file (Condition) 87
- xcl:enter-debugger-p (Function) 119
- xcl:find-proceed-case (Function) 81
- xcl:global (declaration specifier) 13
- xcl:handler-bind (Macro) 65,72
- xcl:ignore-errors (Macro) 74
- xcl:illegal-go (Condition) 86
- xcl:illegal-return (Condition) 86
- xcl:illegal-throw (Condition) 86
- xcl:invoke-proceed-case (Function) 81
- xcl:make-condition (Macro) 69

-
- xcl:open-stream-p (*Function*) 57
 - xcl:proceed (*Function*) 82
 - xcl:proceed-case (*Macro*) 65,75
 - xcl:proceed-case (*Type*) 83
 - xcl:proceed-case-name (*Function*) 80
 - xcl:read-error (*Condition*) 87
 - xcl:readvise-function (*Function*) 109
 - xcl:rebreak-function (*Function*) 102
 - xcl:serious-condition (*Condition*) 85
 - xcl:signal (*Function*) 70
 - xcl:simple-condition (*Condition*) 85
 - xcl:simple-error (*Condition*) 86
 - xcl:simple-warning (*Condition*) 86
 - xcl:stack-overflow (*Condition*) 86
 - xcl:storage-condition (*Condition*) 86
 - xcl:store-value (*Function*) 82
 - xcl:stream-error (*Condition*) 87
 - xcl:synonym-stream-p (*Function*) 57
 - xcl:synonym-stream-symbol (*Function*) 57
 - xcl:trace-function (*Function*) 104
 - xcl:two-way-stream-input-stream (*Function*) 58
 - xcl:two-way-stream-output-stream (*Function*) 58
 - xcl:two-way-stream-p (*Function*) 57
 - xcl:unadvise-function (*Function*) 109
 - xcl:unbound-variable (*Condition*) 87
 - xcl:unbreak-function (*Function*) 102
 - xcl:undefined-function (*Condition*) 87
 - xcl:use-value (*Function*) 82
 - xcl:warning (*Condition*) 85
 - Xerox NS Character Code Standard 39
 - Z
 - Zetalisp 85
 - ↑
 - ↑ (debugger command) 115
 -
 - ~* (format directive) 71
 - ~< (format directive) 60
 - `
 - ` (backquote) 59
 - †
 - † (debugger command) 101
 - #
 - #+ (macro character) 59
 - #- (macro character) 59
 - #: (hash colon) 18
 - #< 55
 - #S 55
 - &
 - &body 9-10
 - &environment 7, 10
 - (
 - ((left parentheses) 90
 - *
 - * 61
 - *break-on-warnings* (*Variable*) 72
 - *load-verbose* 61
 - *package* (*Variable*) 17

print-circle (*Variable*) 60

trace-output 105

/

/ 118

:

: (prompt) 112

:abort 58

:default 61

:displaced-to-base 45

:element-type 61

:extendable 45

:fatp 45

:include 54

:index 58

:inline 54

:lap 89

:oldest 61

:package 61

:print-function 55

:read-only-p 45

:type 5

:wild 61

=

= 118

?

?= (debugger command) 116

@

@ (debugger command) 118

**Xerox Artificial Intelligence Systems
250 North Halstead Street
P.O. Box 7018
Pasadena, California 91109-7018**